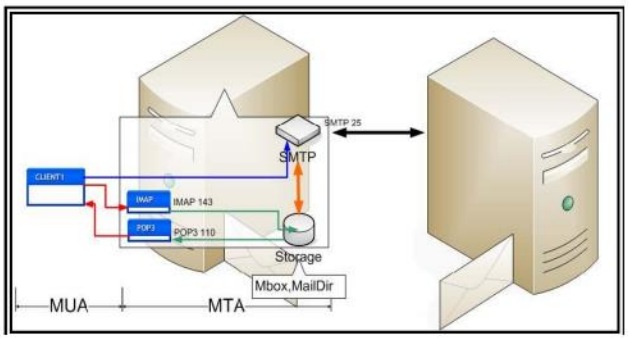
BAB II

LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 EMAIL

Perkembangan teknologi saat ini semakin mempermudah manusia dalam beradaptasi mengikuti perkembangan zaman, teknologi pengiriman pesan kini berkembang pesat dengan adanya teknologi surat elektronik yang disebut *electronic-Mail (E-Mail). Email* merupakan sebuah metode untuk mengirimkan pesan dalam bentuk digital. Pesan ini biasanya dikirimkan melalui medium internet. Sebuah pesan elektronis terdiri dari isi, alamat pengirim, dan alamat-alamat yang dituju.

Sistem e-mail yang beroperasi di atas jaringan berbasis pada model *store and forward.* Sistem ini mengaplikasikan sebuah sistem server *e-mail* yang menerima, meneruskan, mengirimkan, serta menyimpan pesan-pesan user, dimana user hanya perlu untuk mengkoneksikan pc mereka ke dalam jaringan. *E-mail* dapat dianalogikan dengan kotak surat yang ada di kantor POS sedangkan server *e-mail* dapat diibaratkan sebagai kantor POS. Dengan analogi ini sebuah mail server dapat memiliki banyak account *e-mail* yang ada didalamnya[6].



Gambar 2.1 Cara Kerja *Email*

Gambar 2.1 menunjukkan bahwa e-mail yang dikirim belum tentu akan diteruskan ke komputer penerima (*end user*), tapi bisa saja untuk disimpan/dikumpulkan dahulu dalam sebuah komputer server (*host*) yang akan online secara terus menerus (*continue*) dengan adanya media penyimpanan (*storage*) yang relatif lebih besar dibanding komputer biasa. Komputer yang melayani penerimaan *e-mail* secara terus-menerus yang disebut dengan *mailserver* atau *mailhost*[7].

Berikut merupakan ciri-ciri dari email:

a. Pengguna menulis email dan kemudian menginstruksikan aplikasi email untuk mengirimkannya pada alamat penerima.
b. Aplikasi email mengirim pada komputer, mirip seperti seperti kantor pos dan melayani proses penerimaan dan pengiriman email. Perangkat komputer ini disebut *email server*.
c. *Email server* mengidentifikasikan alamat tujuan dan mengirimkannya ke email *server* yang lain yang lebih dekat ke alamat tujuan. Ada kalanya, sebuah email dikirimkan melalui beberapa *email server*, tergantung pada rute yang dilaluinya.
d. Setelah email sampai pada alamat penerima kemudia disimpan di *email server* hingga membuka kotak posnya. Saat penerima membuka kotak posnya, aplikasi email penerima akan meminta email baru yang terdapat di *email server* dan mengunduhnya ke dalam komputer pengguna.
e. Penerima dapat segera membaca email baru yang telah di unduh.

2.1.1 PENTINGNYA PENGELOLAAN EMAIL

Surat merupakan sarana penting bagi suatu instansi dalam bertukar informasi atau kerjasama, pada kebanyakan instansi proses manajemen surat masuk dan surat keluar mulaui dari penerimaan, pembuatan, penyimpanan pendokumentasian hingga verifikasi surat semuanya masih dilakukan secara konvensional.

Dokumentasi surat masuk masih dicatat manual dalam sebuah buku dan tersimpan dalam bentuk *harcopy*, penerapan cara ini menyebabkan menejemen surat menjadi tidak efektif karena untuk membuka surat lama yang tersimpan membutuhkan waktu lama dan perlu dilakukan pencarian satu persatu [8]. Dengan adanya sistem yang mempermudah dokumentasi surat maka akan memberikan beberapa manfaat sebagai berikut:

a. Memberi kemudahan pada admin untuk mengelola keluar masuknya surat.
b. Memberikan kemudahan dalam pencarian dokumen.
c. Pengarsipan secara softcopy yang rapi meminimalisir kerusakan dokumen.

2.2 NAÏVE BAYES

Algoritma Naïve Bayes merupakan sebuah metoda klasifikasi yang menggunakan metode probabilitas dan statistic dikemukakan oleh ilmuwan Inggris Thomas Bayes. Algoritma Naive Bayes memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes.Ciri utama dr Naïve Bayes Classifier ini sangat kuat dan naïf akan independensi dari masing-masing kondisi atau kejadian. Naive Bayes

Classifier bekerja sangat baik dibanding dengan model classifier lainnya. Hal ini dibuktikan pada jurnal [9], mengatakan bahwa "Naïve Bayes Classifier memiliki tingkat akurasi yang lebih baik dibanding model classifier lainnya". Keuntungan penggunan adalah bahwa metoda ini hanya membutuhkan jumlah data pelatihan atau *training data* yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses mengklasifikasikan. Sebagaimana yang diasumsikan sebagai variabel independent, maka dari itu hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan dari keseluruhan dari matriks kovarians. Tahapan dari proses algoritma Naive Bayes adalah:

a. Menghitung jumlah kelas / label.
b. Menghitung Jumlah Kasus Per Kelas.
c. Kalikan Semua Variable Kelas.
d. Bandingkan Hasil Per Kelas

Beberapa keuntungan perhitungan menggunakan naïve bayes adalah mudah diterapkan dan hasil yang baik dengan nilai error rendah dan akurasi tinggi, sementara naïve bayes sendiri memiliki kekurangan jika atribut memiliki keterkaitan yang lemah maka akan mempengaruhi akurasi naïve bayes. Berikut merupakan persamaan yang digunakan dalam perhitungan naïve bayes.

$$P(C|X) = \frac{P\ (C|X)P(c)}{P(X)}$$ naïve bayes (1).

Keterangan dari persamaan naïve bayes 1 :

1. x : Data dengan class yang belum diketahui
2. c : Hipotesis data merupaka suatu class spesifik
3. P(c|x) : Probabilitas hipotesis berdasar kondisi (posteriori probability)

4. P(c) : Probabilitas hipotesis (prior probability)
5. P(x|c) : Probabilita berdasarkan kondisi pada hipotesis
6. P(x) : Probabilitas c

Rumus diatas menerangkan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (Posterior) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut, seringkali disebut prior), dikali dengan peluang kemunculan karakteristik karakteristik sampel pada kelas C (disebut juga likelihood), dibagi dengan peluang kemunculan karakteristik sampel secara global (disebut juga evidence). Karena itu, rumus diatas dapat pula ditulis sebagai berikut :

$$poseterior = \frac{prior\ x\ likehood}{evidence}$$ naïve bayes (2).

Nilai Evidence selalu tetap untuk setiap kelas pada satu sampel. Nilai dari posterior tersebut nantinya akan dibandingkan dengan nilai- nilai posterior kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan. Penjabaran lebih lanjut rumus Bayes tersebut dilakukan dengan menjabarkan (c|x1,…,xn) menggunakan aturan perkalian sebagai berikut :

$$P(c\ |x1, \dots, Xn) = P(C)P(X1, \dots, Xn|C)$$

$$= P(C)P(X1|c)(X2, \dots, Xn|c, X1)$$

$$= P(C)P(X1|c)P(X2|C, X1)(X3, \dots, Xn|C, X1, X2)$$ naïve bayes (3).

Dapat dilihat bahwa hasil penjabaran dari atas tersebut menyebabkan semakin banyak dan semakin kompleksnya factor-faktor syarat yang mempengaruhi nilai probabilitas, yang hampir mustahil untuk dianalisa satu persatu. Akibatnya, perhitungan tersebut menjadi sulit untuk dilakukan. Disinilah digunakan asumsi independensi yang sangat tinggi (naif),

bahwa masing- masing petunjuk saling bebas (independen) satu sama lain. Dengan asumsi tersebut, maka berlaku suatu kesamaan sebagai berikut:

$P(C|X1,..,Xn) = P(C)\prod_{i=1}^{n} P(xi|c)$ rumus naïve bayes (4).

$P(C|X) = P(X_1|C)P(X_2|C) \dots P(X_n|C)P(c)$rumus naïve bayes (5).

Persamaan diatas merupakan model dari *Teorema Naive Bayes* yang selanjutnya akan digunakan dalam proses klasifikasi. Untuk klasifikasi dengan data kontinyu digunakan rumus Densitas Gauss :

$$P = (X_i = X_i \ |Y_i = Y_i) = \frac{1}{\sqrt{2\pi\sigma ij}} e^{-\frac{(x_{i-u_{ij}})2}{2\sigma^2 ij}}$$ naive bayes (6).

Keterangan :

1. P : Peluang
2. Xi : Atribut ke i
3. xi : Nilai atribut ke i
4. Y : Kelas yang dicari
5. yj : Sub kelas Y yang dicari
6. u : Mean, menyatakan rata-rata dari seluruh atribut
7. o : Deviasi standar, menyatakan varian dari seluruh atribut

$$\mu = \frac{1}{n}\sum_{i=i}^{n} x_i$$ rumus mean(7).

$$\sigma = \left[\frac{1}{n-1}\sum_{x=1}^{n}(x_i - \mu)^2\right]^{0.5}$$ rumus standart deviasi(8).

2.3 TEXT MINING

Pada dokumen yang besar dan memiliki skema yang luas digunakan pemberian bobot *term* yang merupakan skema pembobotan atau *Term Weighting* TF-IDF. Kelemahan *scoring* dengan *Jaccard coefficient* adalah tidak

disertakannya frekuensi suatu *term* dalam suatu dokumen, maka diperlukan skoring dengan kombinasi *Term Weighting* TF-IDF [10]. Hal yang perlu diperhatikan dalam pencarian informasi dari koleksi atau kumpulan dokumen yang heterogen adalah pembobotan *term*. *Term* bias seperti kata, frase atau unit hasil *indexing* lainnya dalam suatu dokumen dapat digunakan untuk mengetahui konteks dari dokumen tersebut, maka dari setiap kata tersebut diberikan indikator, yaitu *term weight*. *Term Frequency* adalah frekuensi dari kemunculan sebuah *term* dalam dokumen yang bersangkutan.Jika semakin besar jumlah dari kemunculan suatu *term* dalam dokumen, semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar. Pada *Term Frequency* terdapat beberapa jenis formula yang dapat digunakan :

1. *Biner Term Frequency* yang hanya memperhatikan apakah suatu kata atau term ada atau tidak dalam dokumen, jika ada diberi nilai satu, jika tidak diberi nilai nol.
2. *Raw Term Frequency*, nilai diberikan berdasarkan jumlah kemunculan suatu *term* di dokumen. Contohnya, jika muncul lima kali maka kata tersebut akan bernilai lima.
3. *Logaritmic Term Frequency* hal ini digunakan untuk menghindari dominansi dokumen yang mengandung sedikit *term* dalam *query*, namun mempunyai frekuensi yang tinggi.

2.3.1 PREPROCESSING DATA

Berdasarkan ketidak teraturan struktur data teks, maka proses sistem temu kembali informasi ataupun *text mining* memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih

terstruktur. Salah satu implementasi dari *text mining* adalah *tahap Text Preprocessing*. Tahap *Text Preprocessing* adalah tahapan dimana aplikasi melakukan seleksi data yang akan diproses pada setiap dokumen. Proses preprocessing ini meliputi *case folding, tokenizing, filtering* dan *stemming*.

Tahap preprocessing sebagai berikut :

1. *Case Folding*

Merupakan proses penyamaan keseluruhan teks dalam dokumen. Ini dilakukan untuk mempermudah pencarian. Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh kerena itu peran *case-folding* dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar ysitu menjadikannya huruf kecil [11]. Hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap *delimiter*.

2. *Tokenizing*

Tokenizing merupakan suatu proses penguraian deskripsi yang semula berupa kalimat– kalimat menjadi kata-kata dan menghilangkan delimiter-delimiter seperti tanda titik(.), koma(,), spasidan karakter angka yang ada pada kata tersebut[12]. Namun untuk karakter petik tunggal, titik, semikolon, titk dua atau lainnya dapat memiliki peran yang cukup banyak sebagai pemisah kata. Dalam memperlakukan karakter-karakter dalam teks sangat tergantung pada kontek aplikasi yang dikembangkan. Pekerjaan tokenisasi ini akan semakin sulit jika juga harus memperhatikan struktur bahasa grammatikal.

3. *Filtering*

Ini meruapakan tahap dimana mengambil kata-kata penting dari hasil token. Dengan menggunakan algoritma *stoplist* yang membuang kata kurang penting atau *wordlist* atau menyimpan kata penting. *Stoplist* atau *stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopwords* adalah "yang", "dan", "di", "dari" dan seterusnya. Data *stopword* dapat diambil dari jurna [13]. Kata-kata seperti "dari", "yang", "di", dan "ke" adalah beberapa contoh kata-kata yang berfrekuensi tinggi dan dapat ditemukan hampir dalam setiap dokumen yang disebut sebagai stopword. Penghilangan stopword ini dapat mengurangi ukuran index dan waktu pemrosesan. Selain itu, juga dapat mengurangi level noise.

4. *Stemming*

Digunakan ketika pembuatan indeks dilakukan karena suatu dokumen tidak dapat dikenali langsung oleh suatu *Information Retrieval System* (IRS), oleh karena itu, dokumen tersebut terlebih dahulu perlu dipetakan ke dalam suatu representasi dengan menggunakan teks yang berada di dalamnya. Teknik *Stemming* sangat diperlukan untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen, dan juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk atau form yang berbeda karena mendapatkan imbuhan yang berbeda. Sebagai contoh kata bersama, kebersamaan, menyamai, akan distem ke *root word*-nya yaitu "sama". Namun, seperti halnya *stopping*, kinerja *stemming* juga bervariasi dan sering tergantung pada domain bahasa yang digunakan. Proses *stemming* pada teks berbahasa Indonesia berbeda dengan *stemming* pada teks

berbahasa Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia semua kata imbuhan baik itu sufiks dan prefiks juga dihilangkan.

2.3.2 CONFUSION MATRIX

Pada data mining untuk mengukur atau ada bebeapa cara untuk mengukur kinerja dari model yang dihasilkan salah satunya menggunakan *confusion matriks* (akurasi).*Confusion matrix* adalah suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining. Presisi atau *confidence* adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. *Recall* atau *sensitivity* adalah proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar[14].

Berikut merupakan bentuk dan model perhitungan Confusion matrix :

Tabel 2.1. Bentuk Confusion matrix

Kelas	**terklasifikasi positif**	**terklasifikasi negatif**
Positif	*true postitive (TP)*	*false negative (FN)*
Negative	*false positive (FP)*	*true negative (TN)*

Berdasarkan nilai *True Negative , False Positive, False Negative* dan *True Positive* dapat dihitung dan diperoleh nilai akurasi, presisi dan *recall*. Nilai akurasi ini menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai akurasi dapat diperoleh dengan Persamaan 1. Nilai presisi menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif untuk niali presisi dapat diperoleh dengan

menggunakan persamaan 2. Sementara itu, *recall* menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh sistem. Nilai *recall* dihitung dan dengan persamaan 3.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (1)$$

$$Presisi = \frac{TP}{FP+TP} * 100\% \quad (2)$$

$$Recall = \frac{TP}{FN+TP} * 100\% \quad (3)$$

dimana:

a. TP adalah *True Positive*, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.
b. TN adalah *True Negative*, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.
c. FN adalah *False Negative*, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.
d. FP adalah *False Positive*, yaitu jumlah data positif namun terklasifikasi salah oleh sistem

Sementara itu, pada klasifikasi dengan jumlah keluaran kelas yang lebih dari dua (*multi-class*), cara menghitung akurasi, presisi dan *recall* dapat dilakukan dengan menghitung rata-rata dari nilai akurasi, presisi dan *recall* pada setiap kelas. Persamaan 4, 5, dan 6 merupakan formula untuk menghitung nilai akurasi, presisi dan recall dari sistem klasifikasi *multi-class*.

$$Akurasi = \frac{\sum_{i=1}^{l} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}}{l} * 100\% \quad (4)$$

$$Presisi = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l} (FP_i + TP_i)} * 100\% \quad (5)$$

$$Recall = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l} (TP_i + FN_i)} * 100\% \quad (6)$$

dimana:

a. TP_i adalah *True Positive*, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem untuk kelas ke-i.

b. TN_i adalah *True Negative*, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem untuk kelas ke-i.

c. FN_i adalah *False Negative*, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem untuk kelas ke-i.

d. FP_i adalah *False Positive*, yaitu jumlah data positif namun terklasifikasi salah oleh sistem untuk kelas ke-i

e. l adalah jumlah kelas.

2.4 Tinjauan Pustaka

Berdasarkan penelitian sebelumnya, ada beberapa contoh penelitiam pengklasifikasian menggunakan *naive bayes classifier* yang digunakan untuk pengklasifikasian dokumen diantaranya pengklasifikasian spam mail yang menggunakan metode naive bayes classifier dan itu merupakan pengembangan terbaru dari pemprograman spam filter, metode ini juga memiliki tingkat keakuratan yamh lebih tinggi dibandingkan dengan algoritma sebelumnya contohnya seperti NN Classifier.

Dan adapun penelitian klasifikasi konten berita menggunakan *naïve bayes classifier*.Dalam penelitian ini data yang digunakan berupa berita yang berasal dari beberapa media online. Hasil dari penelitian ini menghasilkan sistem klasifikasi berita berbasis *web* dengan menggunakan bahasa pemrograman PHP dan database MySQL menunjukkan bahwa berita *testing* bias terklasifikasi secara otomatis seluruhnya[15].